Sampling and Meshing Submanifolds in High Dimension

Jean-Daniel Boissonnat, Siargey Kachanovich, Mathijs Wintraecken

▶ To cite this version:

Jean-Daniel Boissonnat, Siargey Kachanovich, Mathijs Wintraecken. Sampling and Meshing Submanifolds in High Dimension. 2019. hal-02386169v2

HAL Id: hal-02386169
https://inria.hal.science/hal-02386169v2

Preprint submitted on 29 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Sampling and Meshing Submanifolds in High Dimension

Jean-Daniel Boissonnat
Université Côte d'Azur, INRIA
[Sophia-Antipolis, France]
jean-daniel.boissonnat@inria.fr

Siargey Kachanovich
Université Côte d'Azur, INRIA
[Sophia-Antipolis, France]
siargey.kachanovich@inria.fr

Mathijs Wintraecken
IST Austria
[Klosterneuburg, Austria]
m.h.m.j.wintraecken@gmail.com

Abstract

This paper presents a rather simple tracing algorithm to sample and mesh an m-dimensional submanifold of $\mathbb{R}^d$ for arbitrary m and d. We extend the work of Dobkin et al. to submanifolds of arbitrary dimension and codimension. The algorithm is practical and has been thoroughly investigated from both theoretical and experimental perspectives. The paper provides a full description and analysis of the data structure and of the tracing algorithm. The main contributions are : 1. We unify and complement the knowledge about Coxeter and Freudenthal-Kuhn triangulations. 2. We introduce an elegant and compact data structure to store Coxeter or Freudenthal-Kuhn triangulations and describe output sensitive algorithms to compute faces and cofaces or any simplex in the triangulation. 3. We present a manifold tracing algorithm based on the above data structure. We provide a detailled complexity analysis along with experimental results that show that the algorithm can handle cases that are far ahead of the state-of-the-art.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Coxeter triangulation, Kuhn triangulation, permutahedron, PL-approximations, isomanifolds

Funding The research leading to these results has received funding from the European Research Council (ERC) under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement No. 339025 GUDHI (Algorithmic Foundations of Geometry Understanding in Higher Dimensions) and the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 754411.

Acknowledgements We thank Dominique Attali (CNRS), Arijit Ghosh (ISI), Vincent Pilaud (École polytechnique), and Aurélien Alvarez (ENS Lyon) for their comments and suggestion. We finally thank all former and current members of Datashape (formerly Geometrica) and Hebert Edelsbrunner's group for the stimulating environment in which the research was conducted.

Lines 500

© Jean-Daniel Boissonnat, Siargey Kachanovich, and Mathijs Wintraecken;
licensed under Creative Commons License CC-BY
35th International Symposium on Computational Geometry (SoCG 2020).
Editors: ???; Article No. 00; pp. 00:1–00:23
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

This paper presents a rather simple algorithm to sample and mesh an m-dimensional submanifold of $\mathbb{R}^d$ for arbitrary m and d. This fundamental problem finds applications in various fields like numerical analysis (to solve nonlinear differential equations) [3], dynamical systems (to approximate invariant manifolds) [39, 28]), chemistry (to study the energy landscape of molecules) [34], robotics (to describe the configuration space of mechanical systems) [31], computer vision and graphics (to visualize time-varying and higher dimensional data) [5, 35].

State-of-the-art. The problem of triangulating differentiable manifolds has a long history in Mathematics dating back to the work of Cairns [12], Whitehead [44] and Whitney[45]. More recently, the problem has received a lot of attention for surfaces of $\mathbb{R}^3$ in the Computational Geometry and Computer Graphics communities. Among the widely used methods are the Marching cube algorithm [42] and Delaunay refinement [14]. In higher dimensions, some early work has been published in Applied Mathematics [1, 22, 39]. A slightly more recent paper by Dobkin et al. [21] attracted the interest of the Computer Graphics community to Coxeter triangulations and their potential use for contour tracing. Although the authors only considered the case of curves ($m = 1$), it was claimed that the method could be "immediately generalized" to submanifolds of higher codimensions. However there has been only few works in that direction. This situation might be explained by the fact that extending the Marching cube algorithm to higher dimensions seems infeasible due to the large number of configurations that should be stored in a lookup table, and that no efficient data structure was known to store and query triangulations in high dimensions. The most recent work we are aware of is the work of Bhaniramka *et al.* [5] which is limited to hypersurfaces ($m = d - 1$) and the work of Min [35]. Min's method applies to submanifolds of any dimension and codimension. It uses an ambient triangulation instead of a cubical grid (the same Freudenthal-Kuhn triangulation used in this paper). However, in their analysis, they consider d as a constant and only report experimental results in dimensions 3 and 4.

The problem we consider is also related to the problem of *manifold reconstruction* from point samples [13, 7, 6]. A major difference however is that in manifold reconstruction, a sample is given as input while here we have to construct the sample using an oracle that queries the manifold. *Manifold sampling* is another related problem which is of fundamental algorithmic significance in statistics. Yet, not much is known beyond the convex case and the case of hypersurfaces ($m = d - 1$) [20, 36].

Contributions. This paper is the first of a series of related papers to fill the gap. In this paper, we extend the work of Dobkin et al. [21] to submanifolds of arbitrary codimension. The algorithm is practical and has been thoroughly investigated from theoretical and experimental perspectives. This paper provides a full description and analysis of the data structure and of the tracing algorithm.

Guarantees on output of the the algorithm are established in two companion papers, one for the case of isomanifolds [10] and one for general smooth submanifolds [8]. Specifically, for isomanifolds we prove [10] that the output is a PL-manifold that has the same topology type as $\mathcal{M}$, and whose Fréchet distance to $\mathcal{M}$ is small. Implementation details and experimental results will be discussed in [9]. The case of submanifolds with boundaries and, more generally, of stratified manifolds can be handled in very much the same way [9, 10].

The content of this paper is as follows. In Section 2, we first discuss Freudenthal-Kuhn and Coxeter triangulations, the latter exclusively of type $\tilde{A}_d$. These triangulations have different origins. Coxeter triangulations derive from geometric group theory, in particular affine Weyl groups. Freudenthal-Kuhn triangulations are combinatorial by nature. Both triangulations are the same up to a linear transformation, as noted by [21] and fully proved in the appendix. This allows us to use on one hand the nice geometric properties of Coxeter triangulations of type $\tilde{A}_d$, where each simplex is very well shaped (large volume compared to longest edge length) and all simplices are identical up to reflections, and on the other hand the simple combinatorial definitions of the Freudenthal-Kuhn triangulation. Although most ideas in this section were known prior to this work, they were disseminated in many different areas and difficult to access. We elucidate those ideas, provide full proofs and combine them so as to extend them to arbitrary dimensions when necessary.

We then introduce a new data structure to compactly store Coxeter or Freudenthal-Kuhn triangulations. The data structure is an elegant and efficient representation of the combinatorial structure of those triangulations. We present a point location algorithm and describe output sensitive algorithms for computing faces and cofaces of simplices of all dimensions in the triangulation.

In Section 3, we present a submanifold tracing algorithm based on the above data structure. The algorithm works for smooth submanifolds of any dimension and codimension. Starting from a given seed point, the algorithm probes the manifold using an oracle and outputs both a sample and a PL approximation of the manifold. A distinctive property of our algorithm, when compared to previous methods that work for submanifolds of arbitrary dimension and codimension [1, 35], is that its complexity depends mostly on the intrinsic dimension of the manifold (see Theorem 24 for a precise statement). Furthermore, using dimensionality reduction techniques, we can completely remove the dependency of the result on the ambient dimension (Theorem 26). The algorithm is quite simple and our implementation can handle cases that are far beyond what was possible before (Section 4).

2 Coxeter-Freudhental-Kuhn triangulations

2.1 Permutahedra

We write $[k] = \{1, \ldots, k\}$ and $[k, l] = \{k, \ldots, l\}$.

▶ **Definition 1** (Permutahedron). A *d-permutahedron* is a d-dimensional polytope, which is the convex hull $\mathcal{P}$ of all points in $\mathbb{R}^{d+1}$, the coordinates of which are permutations of $[d+1]$. Formally, this convex hull can be written as: $\mathcal{P} = \mathrm{conv}\{(\sigma(1), \ldots, \sigma(d+1)) \in \mathbb{R}^{d+1} \mid \sigma \in \mathfrak{S}_{d+1}\}$, where $\mathfrak{S}_{d+1}$ denotes the set of permutations of $[d+1]$.

$\mathcal{P}$ is at most d-dimensional since all its vertices lie on the hyperplane of equation $\sum_{i=1}^{d+1} x^i = \frac{d(d+1)}{2}$. Moreover, it can be shown that there are $d+1$ affinely independent vertices in $\mathcal{P}$, proving that $\mathcal{P}$ is exactly d-dimensional (see for example [33, Lemma 3.4]). The facial structure of $\mathcal{P}$ is best described in terms of ordered partitions [46].

▶ **Definition 2** (Ordered partition). Let T be a finite non-empty set, $|T|$ its cardinality, and $l \leq |T|$ a positive integer. An *ordered partition* of T in l parts is an ordered collection of l subsets $\omega = (\omega_1, \ldots, \omega_l)$, such that $\omega_i \subseteq T$ and $\{\omega_1, \ldots, \omega_l\}$ is a partition of T. The ω_i are called the *parts*. We write $OP_l[d]$ for the set of ordered partitions of $[d]$ with l parts and just $OP[d]$ for the set of all ordered partitions of $[d]$.

▶ **Definition 3** (Refinement). Let ω and ϖ be two ordered partitions of $[d+1]$ in k parts and l parts respectively, with $1 \leq k \leq l \leq d+1$. We say that ϖ is a *refinement* of ω if there exist positive integers $a_1, \ldots, a_k$ such that: $(\varpi_1, \ldots, \varpi_{a_1})$ is an ordered partition of ω_1 in a_1 parts, $(\varpi_{a_1+1}, \ldots, \varpi_{a_1+a_2})$ is an ordered partition of ω_2 in a_2 parts, ..., $(\varpi_{a_1+\ldots+a_{k-1}+1}, \ldots, \varpi_{a_1+\ldots+a_k})$ is an ordered partition of ω_k in a_k parts.

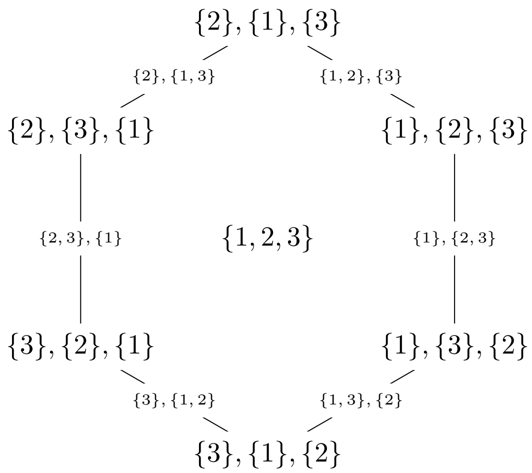

Figure 1 The 2-permutahedron and the ordered partitions associated to its faces.

▶ **Lemma 4** (Facial structure of the permutahedron, Theorem 3.6 of [33]). *The faces of a d-permutahedron are in bijection with the ordered partitions of the set $[d+1]$. More precisely, the l-faces of $\mathcal{P}$ correspond to ordered partitions of $[d+1]$ into $d+1-l$ parts $(\omega_1, \dots, \omega_{d+1-l})$ such that all coordinates in ω_i are smaller than all coordinates in ω_j for $i < j$. If σ and τ are two faces of a d-permutahedron, σ is a subface of τ (noted $\sigma \subseteq \tau$) if and only if the ordered partition associated to σ is a refinement of the ordered partition associated to τ.*

▶ **Corollary 5** (Corollary 3.15 of [33] and Theorem 3 of [37]). *The number of $(d-l)$-dimensional faces in a d-permutahedron is $(l+1)!\, S(d+1, l+1)$, where $S(\cdot,\cdot)$ is the Stirling number of the second kind. It is bounded by $2^{2(d+1)\log(l+1)}$.*

▶ **Corollary 6.** *The number of vertices of a k-face of a d-permutahedron is at most $(k+1)!$*

▶ **Lemma 7.** *The number of facets of an l-face σ of a d-permutahedron is $O\left(2^l\right)$.*

Proofs of the previous two corollaries are added in Appendix B for completeness.

2.2 Freudhental-Kuhn triangulation

The Freudhental-Kuhn (FK for short) triangulation is obtained from the d-grid, i.e. the unit cubical tessellation of $\mathbb{R}^d$ that consists of copies of the unit d-cube along the integer lattice $\mathbb{Z}^d$. By triangulating each d-cube in the grid in an appropriate way to be described now, we obtain the FK-triangulation of $\mathbb{R}^d$.

▶ **Definition 8.** Let $x \in \mathbb{R}^d$ and write $z^i = x^i - \lfloor x^i \rfloor$. We denote by $e_1, \dots, e_d$ the basis vectors and introduce, for reasons that will be clear later, the extra vector $e_{d+1} = -\sum_{i=1}^d e_i$. We introduce the convention that $z^{d+1} = 0$. We associate to x the ordered partition $\omega = (\omega_1, \dots, \omega_{l+1})$ of $[d+1]$ where the ω_i are obtained by sorting the z^i in decreasing order. Specifically, with $\omega_i = \{\omega_i(1), \dots, \omega_i(m_i)\}$, we have

$$1 > z^{\omega_1(1)} = \dots = z^{\omega_1(m_1)} > \dots > z^{\omega_l(1)} = \dots = z^{\omega_l(m_l)} > z^{\omega_{l+1}(1)} = \dots = z^{\omega_{l+1}(m_{l+1})} = 0. \tag{1}$$

▶ **Lemma 9.** *Suppose that $\omega = (\omega_1, \ldots, \omega_{l+1})$ is an ordered partition of $[d+1]$ and let $\sigma = \{v_0, \ldots, v_l\}$ be the l-simplex whose vertices are the points*

$$v_0 = (\lfloor x^1 \rfloor, \ldots, \lfloor x^d \rfloor), \qquad v_i = v_{i-1} + \sum_{i \in \omega_i} e_i \qquad i = 1, \ldots, l. \tag{2}$$

Then x is a point in the relative interior of σ if and only if $z^i = x^i - \lfloor x^i \rfloor$, $i = 1, \ldots, d+1$ (with, as above, $z^{d+1} = 0$ and $d+1 \in \omega_{l+1}$), satisfy (1).

Proof. Because the whole problem is translation invariant, we assume that $v_0 = 0$ without loss of generality, so that the expressions are shorter. Using barycentric coordinates, $z \in \sigma$ can be written as

$$\begin{aligned} z &= \sum_{i=0}^{l} \lambda_i v_i = \sum_{i=0}^{l} \lambda_i \sum_{k=1}^{i} \sum_{j \in \omega_i} e_j \\ &= \lambda_l \left(\sum_{k \in \omega_l} e_k \right) + (\lambda_l + \lambda_{l-1}) \left(\sum_{k \in \omega_{l-1}} e_k \right) + \cdots + (\lambda_l + \cdots + \lambda_1) \left(\sum_{k \in \omega_1} e_k \right), \end{aligned} \tag{3}$$

for some $\lambda_i > 0$ satisfying $\sum_{i=0}^{l} \lambda_i = 1$. We write

$$\begin{aligned} \alpha_{\omega_l(1)} &= \cdots = \alpha_{\omega_l(m_l)} = \lambda_l \\ &\vdots \\ \alpha_{\omega_1(1)} &= \cdots = \alpha_{\omega_1(m_1)} = \lambda_l + \cdots + \lambda_1 \end{aligned} \tag{4}$$

By construction $\alpha_{\omega_i(j)}$ is the $\omega_i(j)$th coordinate of z, denoted by $z^{\omega_i(j)}$, while all coordinates $z^{\omega_{l+1}(1)}, \ldots, z^{\omega_{l+1}(m_{l+1})}$ are zero, because $e_{\omega_{l+1}(i)}$ does not occur in (3), for all i. Moreover, because $\lambda_l + \cdots + \lambda_i > \lambda_l + \cdots + \lambda_{i-1}$, we see that (1) is satisfied.

Conversely, given a point z such that its coordinates satisfy (1), we can read of its barycentric coordinates with respect to the v_i, as defined by (2), from (4). ◀

▶ **Theorem 10.** *The equivalence classes of the points of $\mathbb{R}^d$ with a same ordered partition are the simplices of a triangulation of $\mathbb{R}^d$ called the FK-triangulation.*

Proof. Lemma 9 implies that:

- Any face of a simplex in the FK-triangulation also lies in the FK-triangulation.
- The intersection of two simplices in the FK-triangulation also lie in the FK-triangulation.
- For any point $x \in \mathbb{R}^d$, there is a unique simplex σ such that x lies in the relative interior of σ. Because x has uniquely defined barycentric coordinates with respect to the vertices of σ it is mapped to a unique point in σ.

Hence the partition we have defined is a well-defined triangulation of $\mathbb{R}^d$. ◀

▶ Remark. We note that, by construction, v_0 in Lemma 9 is the smallest vertex of σ in the lexicographical order. Lemma 9 also implies an observation of Freudenthal [25]: all d-simplices in the FK-triangulation can be described by monotone paths along the edges of the cube from vertex $(0, \ldots, 0) + v_0$ to vertex $(1, \ldots, 1) + v_0$. Conversely, any monotone path along the edges of the cubes from $(0, \ldots, 0) + v_0$ to $(1, \ldots, 1) + v_0$ gives a simplex in the FK-triangulation.

Cycles and the permutahedron. This monotone path can be made into a cycle using the extra vector e_{d+1}, introduced by Eaves [22], because by construction $v_0 = v_l + \sum_{i\in\omega_{l+1}} e_i$, with ω as in Definition 8. Because it is a cycle, we can take any vertex of the cycle as a starting point, which means that v_0 no longer has a special role as a starting point of a monotone edge walk. A cycle can now be represented by an ordered partition of $[d+1]$, for which it is not longer necessary that $d+1$ lies in ω_{l+1}, and an (arbitrary) starting point.

We now formalize these general cyclical paths:

▶ **Definition 11** (Permutahedral representation). Let $(v_0,\omega) \in \mathbb{Z}^d \times OP_{l+1}[d]$. To this pair we associate a simplex $\sigma^\omega = \{v_0 = v_0^\omega, v_1^\omega, \dots, v_l^\omega\}$ with

$$v_i^\omega = v_{i-1}^\omega + \sum_{i\in\omega_i} e_i \qquad\qquad i = 1,\dots,l. \tag{5}$$

We say that (v_0,ω) is the permutahedral representation of the simplex σ^ω. If $d+1 \in \omega_{l+1}$ we say that (v_0,ω) is the canonical permutahedral representation of σ^ω. In this case, σ^ω is a simplex in the FK-triangulation in the cube of which v_0 is the minimal vertex with respect to the lexicographical order, as we have seen above. In Lemma 14 and Proposition 15 we'll see that, more generally, $\{(v_0,\omega) \mid \omega \in OP[d+1]\}$ is the star of v_0 in the FK-triangulation, where we identify simplices with their permutahedral representations.

▶ **Definition 12** (Cyclic shifts). Let (v_0,ω) be a permutahedral representation. We define the cyclic shift of (v_0,ω) of length k to the left as (v_0',ω'), where

$$v_0' = v_0 + \sum_{j=1}^{k}\sum_{i\in\omega_j} e_i \qquad\qquad \omega_j' = \omega_{(j+k-1) \bmod (l+1)+1}. \tag{6}$$

Here we use the convention that the sum from 1 to 0 is empty. We write $(v_0',\omega') = (v_0,\omega)\oplus k$.

▶ **Lemma 13.** *The cyclic shift $(v_0',\omega') = (v_0,\omega)\oplus k$ defines the same simplex as (v_0,ω).*

Proof. Follows by inserting (6) in (5). ◀

We now prove that the all permutahedral representations for a fixed v_0, form the star of v_0. This is a crucial property that will be used to efficiently compute faces and cofaces and traverse the triangulation.

▶ **Lemma 14.** *The set $\{(v_0,\omega) \mid \omega \in OP[d+1]\}$, where $OP[d+1]$ is the set of all ordered partitions of $[d+1]$, gives all the simplices in the star of v_0 in FK-triangulation.*

Proof. Let (v_0,ω), with $\omega \in OP_{l+1}[d+1]$, be such that $d+1 \in \omega_k$. Let $(v_0',\omega') = (v_0,\omega)\oplus (l-k+1)$. By Definition 12 and Lemma 13, (v_0,ω) and (v_0',ω') represent the same simplex. Moreover $d+1 \in \omega_{l+1}'$, that is (v_0',ω') is a canonical permutahedral representation. This implies that (v_0',ω') lies in the FK-triangulation by Lemma 9 and Theorem 10.

Conversely, suppose that (v_0',ω') is the canonical permutahedral representation of a simplex in the star of v_0, that is there is some k such that $v_k' = v_0$, with v_k' as in (2). Then $(v_0,\omega) = (v_0',\omega')\oplus k$ is also a permutahedral representation of the same simplex. ◀

Faces. From (5) it is clear that merging two consecutive parts in the ordered partition $\omega = (\omega_1,\dots,\omega_{l+1})$ corresponds to removing a vertex from the simplex, that is taking a facet. Here we stress that we allow to merge ω_1, and ω_{l+1}, but in that case we have to change the base point of the cycle to $v_0 + \sum_{l\in\omega_1} e_l$. For example, when looking at the two dimensional example in Figure 2, we see that the edges that contain y in the red triangle with permutahedral representation $(y, (\{1\}, \{2\}, \{3\}))$ are $(y, (\{1, 2\}, \{3\}))$, and $(y, (\{1\}, \{2, 3\}))$. The third edge of the red triangle is $(y', (\{2\}, \{1, 3\}))$. Generally, given an ordered partition ω in $l+1$ parts all $(l-j)$-faces can be found by merging j consecutive parts in ω (for example merging ω_1 with ω_2 and ω_3 with ω_4), where we allow ω_{l+1} to merge with ω_1, but in this case we again need to change the base point.

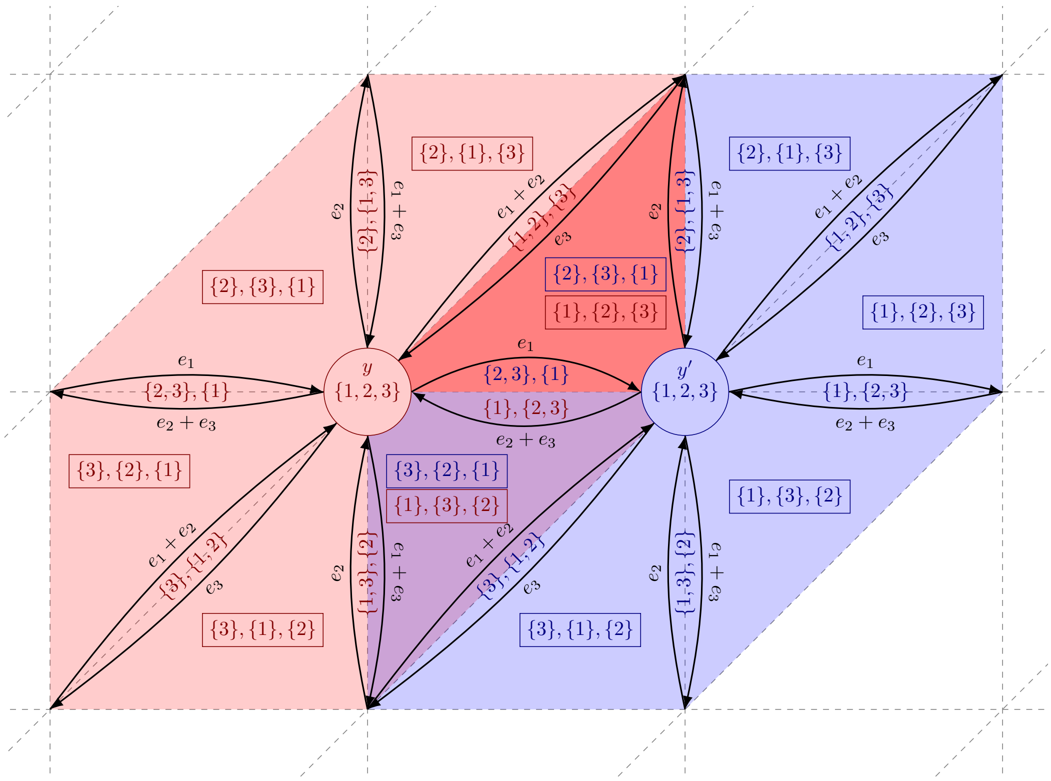

Figure 2 The permutahedral representation of the simplices in the stars of vertices y and y'.

Because the combinatorial structure of the faces is compatible with the permutahedron, Lemma 14 immediately gives:

▶ **Proposition 15.** *The star of v_0 is dual to a permutahedron (combinatorially).*

This proposition also explains the nomenclature permutahedral representation.

2.3 Basic operations

Point location Given a point $x \in \mathbb{R}^d$ Lemma 9 tells us how to find the canonical permutahedral representation of the simplex in which x is contained. The complexity of point location is dominated by the sorting of the $z^i = x^i - \lfloor x^i \rfloor$, which takes $O(d \log d)$ time and requires $O(d)$ space.

Face computation. Let σ be an l-simplex whose canonical permutahedral representation is (v_0, ω), where ω is an ordered partition of $[d+1]$ into $l+1$ parts. The computation of all k-faces of σ goes as follows. We use Ehrlich's subset generation algorithm [24] to compute all the subsets of $k+1$ elements from $\{v_0, \ldots, v_l\}$. Let $\tau = \{v_{m_0}, \ldots, v_{m_k}\}$ be such a subset. τ is a k-face of σ. We then compute the canonical permutahedral representation of all those k-faces τ.

We first sort the m_i so that $m_0 < \cdots < m_k$ using counting sort. Then, the canonical permutahedral representation $(\tilde{v}_0', \omega')$ of τ is found by merging consecutive parts of ω so as to obtain $k+1$ parts as follows :

$$v_0' = v_{m_0} = v_0 + \sum_{j \in \omega_1} e_j + \ldots + \sum_{j \in \omega_{m_0-1}} e_j$$

$$\omega_i' = \omega_{m_{i-1}} \cup \ldots \cup \omega_{m_i - 1} \qquad \text{for } i \in \{1, \ldots, k\}$$

$$\omega_{k+1}' = (\omega_1 \cup \ldots \cup \omega_{m_0-1}) \cup (\omega_{m_k} \cup \ldots \cup \omega_{l+1}).$$

The complexity of computing all subsets of $k+1$ vertices of σ using Ehrlich's algorithm takes time $O(k+s)$ where $s = \binom{l+1}{k+1}$ is the number of subsets. Computing, for each such k-simplex its permutahedral representation takes $O(d)$ time.

► **Lemma 16.** *Let σ be an l-simplex in the FK-triangulation of $\mathbb{R}^d$ given by its canonical permutahedral representation. Computing the canonical permutahedral representations of all its k-faces can be done in time $O(ds)$, where $s = \binom{l+1}{k+1}$ is the number of k-faces of an l simplex. The space complexity of the algorithm is $O(l)$ from the counting sort.*

Coface computation. Computing the faces of a simplex σ consisted in coarsifying its ordered partition. The computation of cofaces is the reverse. Here we refine the ordered partition. Specifically, if σ is a k-simplex represented by its canonical permutahedral representation (v_0, ω), and we want to compute its l-cofaces, we need to compute all refinements of ω into $l+1$ parts.

More precisely, we need to subdivide each ω_i in $a_i \leq |\omega_i|$ subparts so that $\sum_{i=1}^{k+1} a_i = l+1$. This can be done in time proportional to the number $k+1$ of the generated subparts. We then need to consider all the permutations of these subparts since we are interested in ordered partitions. Using known algorithms by Walsh [41] and Ruskey and Savage [38], we can compute all the ordered partitions associated to the l-cofaces of σ in time proportional to the number of such cofaces. We thus obtain all the permutahedral representations (v_0, ω') of all the l-cofaces of σ.

It is important to notice that all cofaces of σ have v_0 as a vertex. However v_0 is not necessarily the minimal vertex of some of the computed cofaces. We thus have to identify the minimal vertex of each computed coface and use cyclic shifts as in Lemma 14 to obtain the canonical permutahedral representation of the coface.

► **Lemma 17.** *Let σ be a k-simplex in the FK-triangulation of $\mathbb{R}^d$ given by its permutahedral representation. Computing the permutahedral representations of all its l-cofaces can be done in time $O(ds)$, where s is the number of l-cofaces of a k-simplex in the FK-triangulation. The space complexity of the algorithm is $O(d)$.*

2.4 Coxeter triangulations of type $\tilde{A}_d$

The Freudenthal-Kuhn triangulation is closely related to the Coxeter triangulation [18] of type $\tilde{A}_d$. There are many equivalent ways to define the Coxeter triangulation of type $\tilde{A}_d$, see [11, 15, 29]. We recall the following:

▶ **Definition 18.** Let $P = \{(x^i) \in \mathbb{R}^{d+1} \mid \sum_i x^i = 0\}$ and consider the d-simplex with vertices u_k in P.

$$u_0 = \left(0^{\{d+1\}}\right) \qquad u_k = \left(\left(-\frac{d+1-k}{d+1}\right)^{\{k\}}, \left(\frac{k}{d+1}\right)^{\{d+1-k\}}\right), \qquad k \in [d],$$

where $x^{\{k\}}$ denotes k consecutive coordinates x. The Coxeter triangulation of type $\tilde{A}_d$ in P is found by consecutively reflecting the simplex in its faces.

The following lemma relates Coxeter and FK-triangulations and was first stated in [21].

▶ **Lemma 19.** *The Freudenthal-Kuhn triangulation and Coxeter triangulation of type $\tilde{A}_d$ are identical up to a linear transformation.*

A proof can be found in Appendix A. We will now call any triangulation of Euclidean space that is the image of a Coxeter triangulation under a non-degenerate affine map a Coxeter-Freudenthal-Kuhn triangulation, or CFK-triangulation for short. Moreover, we have

▶ **Lemma 20** ([15])**.** *The Coxeter triangulation of type $\tilde{A}_d$ is a Delaunay triangulation.*

We note that Proposition 15, together with Lemmas 19 and 20, give an alternative self-contained proof of the known fact [17, Chapter 21, Section 3.F] (also proved in Appendix A for completeness) that the Voronoi cell of a vertex in $\tilde{A}_d$ is a (combinatorial) permutahedron.

The simplices in the Coxeter triangulation of type $\tilde{A}_d$ have extremely good quality [15]. For example, the volume compared to the longest edge length to the d-th power is large. As we will see, the exceptional quality of Coxeter improves the running of our algorithms.

2.5 Data structure for storing CFK-triangulations

To store an ambient CFK-triangulation for the manifold tracing algorithm in Section 3, we use the following data structure. This data structure contains information on both the combinatorial structure and the geometry of the triangulation. The combinatorics of the triangulation is given through the canonical permutahedral representation of its simplices and the algorithms from Section 2.2. The geometry of the triangulation is specified by the affine transformation that maps the FK-triangulation of $\mathbb{R}^d$ to the CFK-triangulation. The affine transformation is given by a $d \times d$ matrix Λ and a d-vector b.

▶ Remark. Matrix Λ is used to compute the coordinates of the vertices of simplices and, for the most useful cases in practice, needs not to be explicitly stored. For the FK-triangulation, Λ is the identity matrix and $b = 0$, therefore no storage is required. For the Coxeter triangulation of type $\tilde{A}_d$, we can directly access the coordinates of vertices as given in Definition 18.

3 Sampling and meshing submanifolds

In this section, we describe an algorithm that will compute a PL-approximation of an m-submanifold of $\mathbb{R}^d$ for arbitrary d and $m \leq d$. The algorithm can be considered as an alternative to the Marching Cube algorithm [32] where the usual cubical grid is replaced by a CFK — preferably the Coxeter — triangulation of the ambient space. Taking a triangulation instead of a grid is a major advantage in high dimensions that has been recognized in the pioneering works of Allgower and Schmidt [2] and of Dobkin et al. [21]. See also [35]. By taking as a triangulation of the ambient space a CFK-triangulation, we keep two main advantages of using grids: very limited storage and fast basic operations.

Algorithm 1 Manifold tracing algorithm

input : Triangulation $\mathcal{T}$ of $\mathbb{R}^d$, manifold $\mathcal{M}$ of dimension m, seed point $x_0 \in \mathcal{M}$
output: Set $\mathcal{S}$ of the simplices in $\mathcal{T}$ of dimension $k = d - m$ that intersect $\mathcal{M}$
Translate $\mathcal{T}$ so that x_0 coincides with the barycentre of a k-dimensional face τ_0 in $\mathcal{T}$
Initialize the queue $\mathcal{Q}$ and the set $\mathcal{S}$ with τ_0
while *the queue* $\mathcal{Q}$ *is not empty* **do**
- Pop a k-dimensional simplex τ from $\mathcal{Q}$
- **foreach** *cofacet* ϕ *of* τ **do**
  - **foreach** *facet* ρ *of* ϕ **do**
    - **if** ρ *does not lie in* $\mathcal{S}$ *and intersects* $\mathcal{M}$ **then**
      - Insert ρ to the queue $\mathcal{Q}$
      - Insert ρ together with the intersection point to the output set $\mathcal{S}$

3.1 Manifold tracing algorithm

Let $\mathcal{M}$ be an m-dimensional compact submanifold of the Euclidean space $\mathbb{R}^d$. Both m and d are known but arbitrary and will be considered as parameters in the complexity analysis. The algorithm will use a CFK-triangulation $\mathcal{T}$ of $\mathbb{R}^d$, which is stored using the data structure from Section 2.5. We assume that the manifold $\mathcal{M}$ and the triangulation $\mathcal{T}$ satisfy a genericity hypothesis:

▶ **Hypothesis 21** (Genericity). *The manifold has an empty intersection with all simplices of dimensions strictly lower than k in the triangulation $\mathcal{T}$. The intersection of the manifold $\mathcal{M}$ and any k-dimensional simplex in the triangulation $\mathcal{T}$ is a single point.*

▶ Remark. It turns out [10] that for an isomanifold $f^{-1}(0)$ it suffices to find the intersection points of $f_{\mathrm{PL}}^{-1}(0)$ with the k-simplices under very weak conditions. Here f_{PL} denotes the function that is linear on every simplex in $\mathcal{T}$ and coincides with f on the vertices of $\mathcal{T}$. We stress that $f_{\mathrm{PL}}^{-1}(0)$ satisfies the genericity hypothesis with probability one.

We assume that we know a point on the manifold $x_0 \in \mathcal{M}$, from which the algorithm starts. If $\mathcal{M}$ consists of multiple connected components, then a seed point per each connected component must be provided and we proceed in the same manner for each component. So we will assume for now that $\mathcal{M}$ is connected.

In addition, we assume that the manifold $\mathcal{M}$ can be accessed through an *oracle* that allows us to answer whether a k-simplex in the triangulation $\mathcal{T}$ intersects the manifold $\mathcal{M}$. Here, $k = d - m$ is the codimension of $\mathcal{M}$. In the following, we will refer to this oracle as the *intersection oracle*.

The algorithm is described as Algorithm 1. We first translate the coordinate frame so that x_0 is the barycenter of a k-simplex of $\mathcal{T}$ (any such simplex is fine). This simplex is put in the set $\mathcal{S}$ of the simplices in $\mathcal{T}$ of dimension $k = d - m$ that intersect $\mathcal{M}$. Then, given such a simplex, we look at all its cofacets that have not been considered yet and consider all the facets of those cofacets that have not been considered yet. This can be done using a queue $\mathcal{Q}$ of simplices to consider. Each of these simplices is queried with the intersection oracle and, if it is found to intersect $\mathcal{M}$, it is added to $\mathcal{S}$. Upon termination, $\mathcal{S}$ contains all the k-dimensional simplices of $\mathcal{T}$ that intersect $\mathcal{M}$. Since, by our genericity assumption, each k-simplex in $\mathcal{S}$ intersects $\mathcal{M}$ in a single point, $|\mathcal{S}|$ is also the size of the sample produced by our algorithm. A better approximation of the sample is of course possible if we have at our disposal a more powerful intersection oracle that not only detects intersections but can also compute intersection points between the simplices in $\mathcal{S}$ and $\mathcal{M}$.

A polyhedron can be deduced from $\mathcal{S}$ by taking the dual faces of the simplices in $\mathcal{S}$. A more precise approximation can be obtained if, in addition to the intersection oracle, we can also compute the intersection points $\mathcal{S} \cap \mathcal{M}$. This will be described in full detail in a companion paper.

3.2 Complexity analysis

We can easily bound the complexity of the manifold tracing algorithm as a function of the size of the output.

▶ **Proposition 22.** *The time complexity of the algorithm is $O\left(k2^m I|\mathcal{S}|\right)$. where I is the time complexity of one call of the intersection oracle and $|\mathcal{S}|$ is the size of the output.*

Proof. The complexity of the initialization is $O(d)$. The complexity of each iteration of the while loop consists of: computing the cofacets of the popped k-dimensional simplex in the queue, computing facets of these cofacets and applying the intersection oracle on each of these facets. From Lemma 7, the number of cofacets is $O(2^m)$. Each of these cofacets has $k+2$ facets. Therefore, for each iteration of the while loop, the algorithm applies the intersection oracle on $O(k2^m)$ simplices. By using this observation and the complexities in Lemmas 16 and 17, the total time complexity of each iteration of the while loop follows:

$$O(d2^m) + O(dk2^m) + O(k2^m I) = O(k2^m(d+I)) = O(k2^m I).$$

Since there are $|\mathcal{S}|$ iterations of the while loop, the result follows. ◀

We will now express the size of the output in terms of quantities that depend on the manifold and the resolution of the triangulation.

▶ **Proposition 23** (Size of the output)**.**

$$|\mathcal{S}| \leq \frac{C}{\Theta\sqrt{m}} \left(\frac{2\pi e}{k}\right)^{k/2} \left(\frac{2}{\delta}\right)^m \mathrm{vol}_m(\mathcal{M}) \quad = \quad O\left(2^{O(d\log d)}\, \frac{\mathrm{vol}_m(\mathcal{M})}{\delta^m}\right).$$

where:

- $\mathrm{vol}_m(\mathcal{M})$ *is the m-dimensional volume of $\mathcal{M}$,*
- *δ is the diameter of the d-simplices of $\mathcal{T}$ and a measure of the resolution of $\mathcal{T}$,*
- *V is the volume of any d-simplex of $\mathcal{T}$, and $\Theta = \frac{V}{\delta^d}$ its fatness,*
- *C is a constant that does not depend on d, m or δ.*

Proof. Let $\mathcal{N}$ be the set of the d-dimensional cofaces of the simplices in $\mathcal{S}$, and let N be the cardinality of $\mathcal{N}$. In the proof we will use constants C_1, C_2, C_3 that are constants that do not depend on d, m nor δ.

Upper bound on N. Write $\mathcal{M}^\delta$ for the tubular neighbourhood of $\mathcal{M}$ of radius δ, i.e. the set of points at distance at most δ from $\mathcal{M}$. Since the d-dimensional simplices in $\mathcal{N}$ have pairwise disjoint interiors and all lie inside $\mathcal{M}^\delta$, we have

$$N \cdot V \leq \mathrm{vol}_d(\mathcal{M}^\delta). \tag{7}$$

According to the tube formula of Weyl [43, 26] and writing B_k for the volume of the unit ball of dimension k, there exists constants C_1 and C_2 such that

$$\mathrm{vol}_d(\mathcal{M}^\delta) \le C_1 B_k \delta^k \,\mathrm{vol}_m(\mathcal{M}) \le C_2 \left(\frac{2\pi e}{k}\right)^{k/2} \delta^k \,\mathrm{vol}_m(\mathcal{M}) \tag{8}$$

By combining the two inequalities (7) and (8), we get:

$$N \le \frac{\mathrm{vol}_d(\mathcal{M}^\delta)}{V} \le C_2 \left(\frac{2\pi e}{k}\right)^{k/2} \frac{\delta^k}{V} \,\mathrm{vol}_m(\mathcal{M}) \le C_2 \left(\frac{2\pi e}{k}\right)^{k/2} \frac{1}{\Theta} \frac{\mathrm{vol}_m(\mathcal{M})}{\delta^m}. \tag{9}$$

where Θ is the fatness of the simplices in the triangulation $\mathcal{T}$. Note that the dependency of N on $1/\delta$ is exponential in m but not in d.

Upper bound on $|\mathcal{S}|$. Now, we express $|\mathcal{S}|$ in terms of N, d and m. For this, we count the number INC of incidences of the k-dimensional simplices in $\mathcal{S}$ and the d-dimensional simplices in $\mathcal{N}$ in two ways:

$$\sum_{\tau\in\mathcal{S}} |\mathrm{cof}(\tau, d)| = \mathrm{INC} = \sum_{\sigma\in\mathcal{N}} |\mathrm{fac}(\sigma, k) \cap \mathcal{S}|. \tag{10}$$

The number of d-cofaces of a k-simplex is given by Corollary 6 applied to the dual Voronoi face of τ. Hence there exists a constant C_3 such that for any k-dimensional simplex $\tau \in \mathcal{S}$, we have:

$$|\mathrm{cof}(\tau, d)| \ge C_3 \left(\frac{d}{m+1}\right)^{m+1}.$$

On the other hand, for each d-dimensional simplex $\sigma \in \mathcal{N}$, we have :

$$|\mathrm{fac}(\sigma, k) \cap \mathcal{S}| \le |\mathrm{fac}(\sigma, k)| = \binom{d+1}{m+1}.$$

Equation (10) becomes

$$C_3 \left(\frac{d}{m+1}\right)^{m+1} |\mathcal{S}| \le \sum_{\tau\in\mathcal{S}} |\mathrm{cof}(\tau, d)| = \sum_{\sigma\in\mathcal{N}} |\mathrm{fac}(\sigma, d) \cap \mathcal{S}| \le \binom{d+1}{m+1} N,$$

from which we get using (9) that there exists a constant C such that

$$|\mathcal{S}| \le C \left(\frac{2\pi e}{k}\right)^{k/2} \frac{1}{\Theta} \mathrm{vol}_m(\mathcal{M})\, m^{-1/2} (2/\delta)^m.$$

Bound on fatness. The fatness term Θ in the expression in Proposition 23 depends on the choice of the triangulation $\mathcal{T}$. The fatness Θ_{CT} of the d-dimensional simplices in the Coxeter triangulation is given by (see [15])

$$\frac{1}{\Theta_{CT}} = O\left(\frac{d^{(d+1)/2}\, d!}{2^d}\right) = O(2^{O(d \log d)}). \tag{11}$$

while the fatness Θ_{FKT} of the d-dimensional simplices in the FK-triangulation of $\mathbb{R}^d$ is given by (see [19])

$$\frac{1}{\Theta_{FKT}} = O\left(d^{d/2}\, d!\right) = O(2^{O(d \log d)}). \tag{12}$$

Note that, in both cases, Θ depends on the ambient dimension but not on δ. Note also that, while similar, the two bounds on the fatness differ by a factor $2^d/\sqrt{d}$. We thus expect that using the Coxeter triangulation as the ambient triangulation will give a smaller output $\mathcal{S}$ than the one we obtain using the Freudenthal-Kuhn triangulation. This is confirmed in practice as shown in Section 4 (see Figure 6). ◀

We combine Propositions 22 and 23 in the following theorem.

▶ **Theorem 24.** *The time complexity of the manifold tracing algorithm is* $2^{O(d \log d)}\, I\, \frac{\mathrm{vol}_m(\mathcal{M})}{\delta^m}$, *where I is the time complexity of one call of the intersection oracle.*

Cost of the oracle. The cost of I depends on how the submanifold is given. As an example, consider the case where $\mathcal{M}$ is the PL-approximation of the zero set of a function $f : \mathbb{R}^d \to \mathbb{R}^k$ and assume that evaluating f at any point $x \in \mathbb{R}^d$ can be done in time polynomial in d (which, in particular, is true if each f^i is a polynomial in the coordinates of x). Then I depends polynomially on d too. Indeed, consider a k-simplex σ of the triangulation on which we call the intersection oracle and let H denote the m-flat that linearly interpolates $f^{-1}(0)$ inside σ. To implement the oracle, we first evaluate f at the vertices of σ. We then compute the barycentric coordinates of the (generically unique) point of intersection of the affine hull of σ with H. Lastly, we check whether the barycentric coordinates are all non-negative (to ensure that the intersection point lies inside σ). It follows that the cost of the oracle is the cost of evaluating f at the $k+1$ vertices of σ plus the cost of solving a linear system of k equations and k unknowns, which can be done in time $O(k^{2.375})$.

Dimensionality reduction. As seen from Proposition 23, the size $\mathcal{S}$ of the output of the algorithm, considered as a function of the resolution $1/\delta$ of the triangulation, depends exponentially on m (which is to be expected) and not on d (which is fortunate). Nevertheless, the size of the output depends exponentially on d. This, in particular, means that the sample constructed by the algorithm, although δ-dense, is not guaranteed to be $\mu\delta$ separated for some constant μ. In other words, the output sample is not a net of the manifold.

We can improve on the bound on $\mathcal{S}$ by using dimensionality reduction techniques and, specifically, a variant of the celebrated Johnson-Lindenstrauss lemma for manifolds. We depart from our previous worst-case analysis by allowing some approximation factor ε and tolerate a guarantee that holds only with high probability.

▶ **Theorem 25** (Johnson-Lindenstrauss lemma for manifolds [16, 40])**.** *Pick any $\varepsilon, \eta > 0$, and let $d' = \Omega\left(\frac{m}{\varepsilon^2}\log\frac{1}{\varepsilon} + \frac{1}{\varepsilon^2}\log\frac{\Gamma}{\delta}\right)$, where Γ is a quantity that depends only on intrinsic properties of $\mathcal{M}$. Let Φ be a random affine subspace of dimension d'. Then, with probability $> 1-\eta$, for all $x, y \in \mathcal{M}$*

$$(1-\varepsilon)\sqrt{\frac{d'}{d}} \leq \frac{\|\Phi x - \Phi y\|}{\|x-y\|} \leq (1+\varepsilon)\sqrt{\frac{d'}{d}}.$$

It follows that the image $\Phi\mathcal{M}$ of $\mathcal{M}$ will be a submanifold of dimension m embedded in $\mathbb{R}^{d'}$. One can now run the manifold tracing algorithm in $\mathbb{R}^{d'}$ to sample and mesh $\Phi\mathcal{M}$. The algorithm works as described before except that we need another oracle that, given a $(d'-m)$-simplex σ of the CFK-triangulation of $\mathbb{R}^{d'}$, decides whether its inverse image $\Phi^{-1}\sigma$, which is a $(d-d')$-dimensional flat strip in $\mathbb{R}^d$, intersects $\mathcal{M}$ or not.

Due to the scaling factor $\sqrt{d/d'}$, the resolution of the triangulation in the low dimensional plane has to be scaled by the same factor if one wants to satisfy a given sampling density on $\mathcal{M}$. Since the geometry of the manifold (reach and volume) is also scaled in the same way [23], the analysis of the algorithm will be unchanged. Theorem 24 shows that the

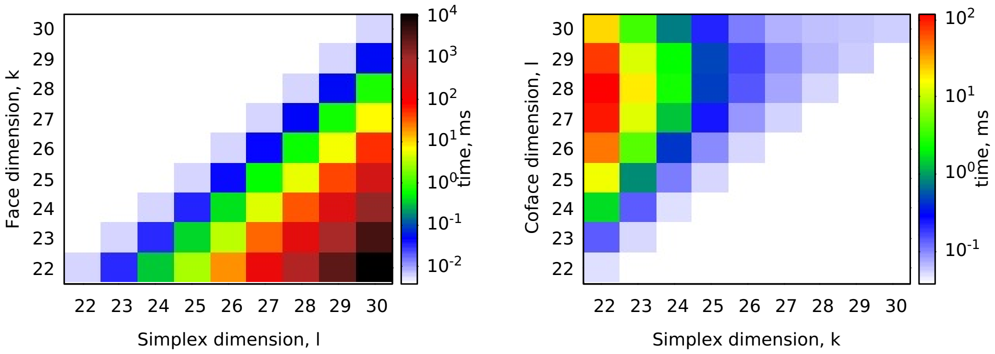

Figure 3 On the left: comparison of the execution time of the face and the coface generation algorithm for simplices of various dimensions in a CFK-triangulation of $\mathbb{R}^{30}$. Because the average computation time of a face or coface is constant, the presented time is proportional to the number of faces or cofaces of respective simplices.

output sample will have size $O\left(2^{O(d' \log d')} \frac{\mathrm{vol}_m(\mathcal{M})}{\delta^m}\right)$. Since d' does not depend on the ambient dimension d by Theorem 25, neither does the size of the output sample.

▶ **Theorem 26.** *Pick any $\varepsilon, \eta > 0$, and let $d' = \Omega\left(\frac{m}{\varepsilon^2} \log \frac{1}{\varepsilon} + \frac{1}{\varepsilon^2} \log \frac{\Gamma}{\delta}\right)$, where Γ is a quantity that depends only on intrinsic properties of $\mathcal{M}$. Let Φ be a random affine subspace of dimension d'. Then, with probability $> 1-\eta$, we can sample and mesh $\mathcal{M}$ using the tracing algorithm in $\mathbb{R}^{d'}$ and the new oracle. The size of the output is $O\left(2^{O(d' \log d')} \frac{\mathrm{vol}_m(\mathcal{M})}{\delta^m}\right)$.*

The previous theorem bounds the size of the output. The complexity of the new oracle is the same as the complexity of the basic intersection oracle.

4 Experimental results

The data structure of Section 2 and the algorithm of Section 3 have been implemented in C++. The code is robust and fast, and will be released soon in the GUDHI library [27]. Full detail on the implementation, including the implementation of the oracle, will be reported in a companion paper together with experimental results [9]. See also [30].

In this section, we explore the dependency of our C++ implementation of the data structure for the ambient CFK-triangulation and of the manifold tracing algorithm on the properties of the triangulation and of the input manifold.

Data structure.

In Figure 3, we present the time of generating all faces (on the left) and all cofaces (on the right) of various dimensions of simplices in a CFK-triangulation of $\mathbb{R}^{30}$ using algorithms from Section 2.2. The presented execution time is averaged over 500 tests. Note that both for face and coface generation algorithms, the execution time is proportional to the number of computed elements. On average, these algorithms take time 0.001-0.002 ms per computed face or coface, regardless of the dimensions of the input simplex and of the computed element. In Figure 4, we further illustrate the particular case of facet and cofacet computation, which is essential in the manifold tracing algorithm. We show the dependency of the execution time on two parameters: the ambient dimension d and the codimension m of the input

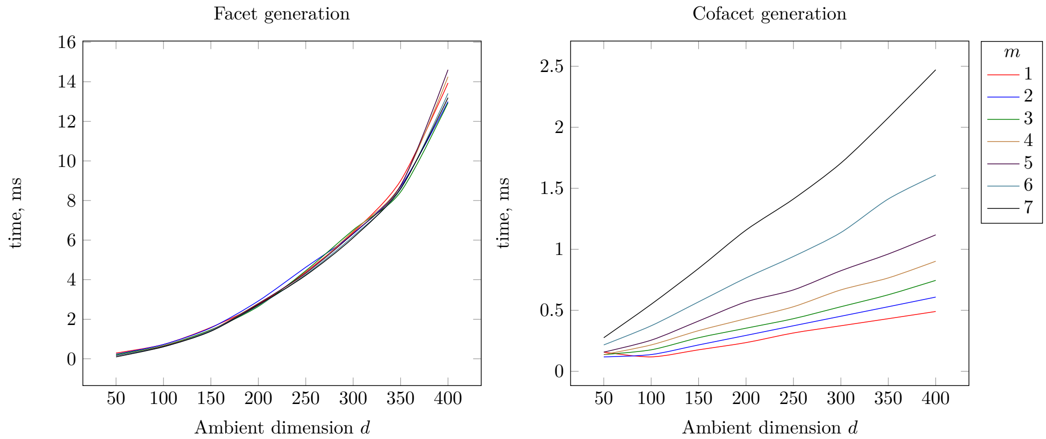

Figure 4 Execution time of the facet and cofacet computation depending on the dimension d of the triangulation and the codimension m of the input simplex.

simplex, which corresponds to the intrinsic dimension of the input manifold in the manifold tracing algorithm.

Manifold tracing algorithm.

We show the performance of our implementation of the manifold tracing algorithm for various ambient and intrinsic dimensions in Figure 5. In Figure 6, we can see that using Coxeter triangulation is beneficial in practice as it produces a smaller output in less time. In Figure 7, we present a PL approximation of a two-dimensional flat torus without boundary embedded in $\mathbb{R}^{10}$ built by the manifold tracing algorithm. The algorithm can be easily adapted to handle submanifolds with boundary. In Figure 8, we present the mesh obtained by our algorithm on a portion of a flat torus embedded in $\mathbb{R}^4$ and cut by a hypersphere. Both surfaces in Figure 7 and 8 are rotated and translated in their respective ambient spaces for visualization purposes. Note that there is no C^2 embedding of the flat torus in $\mathbb{R}^3$.

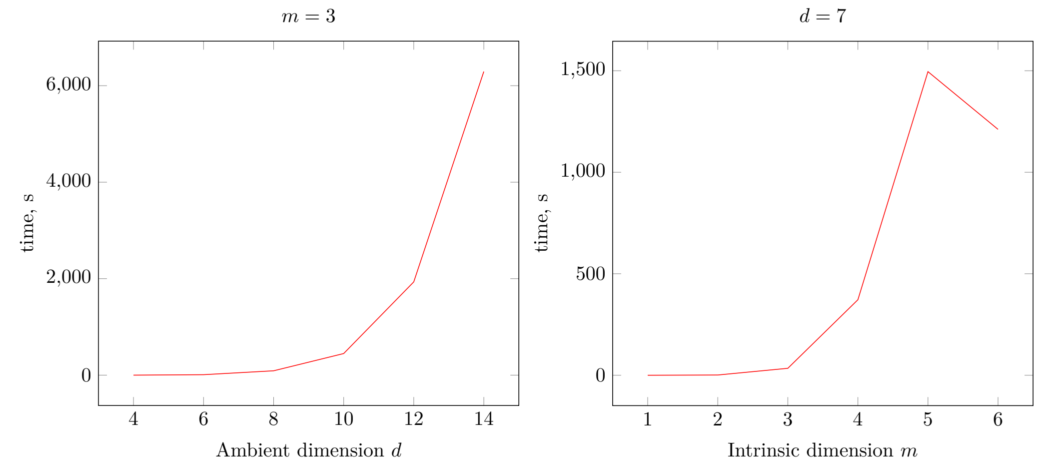

Figure 5 The effect of the ambient dimension d and of the intrinsic dimension m on the computation time of the of the manifold tracing algorithm. The reconstructed manifold in the tests is the m-dimensional sphere embedded in $\mathbb{R}^d$. The ambient triangulation used is a Coxeter triangulation of type $\tilde{A}_d$. The diameter of the full simplices is fixed for all d.

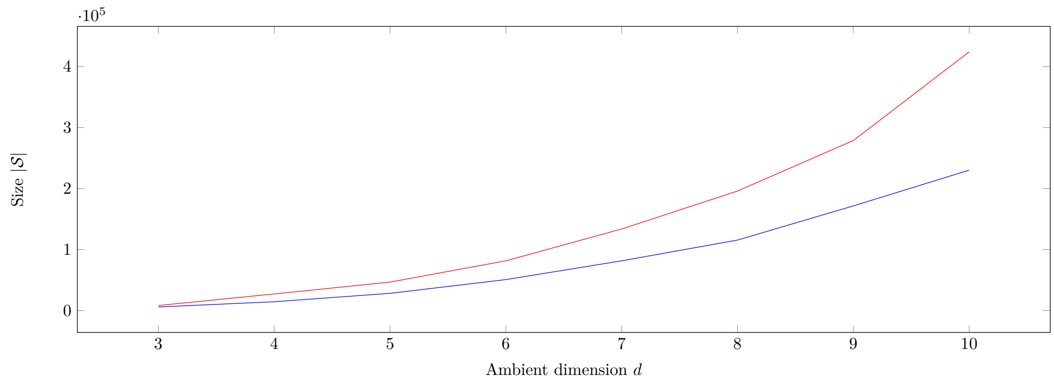

Figure 6 Comparison of the size of the output of the manifold tracing algorithm using two types of the ambient triangulation: a Coxeter triangulation of type $\tilde{A}_d$ (in blue) and the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ (in red) with the same diameter $0.07\sqrt{d}$ of d-dimensional simplices. The reconstructed manifold is the 2-dimensional implicit surface "Chair" embedded in $\mathbb{R}^d$ given by the equations: $(x_1^2 + x_2^2 + x_3^2 - 0.8)^2 - 0.4\left((x_3-1)^2 - 2x_1^2\right)\left((x_3+1)^2 - 2x_2^2\right) = 0$ and $x_i = 0$ for $i > 3$.

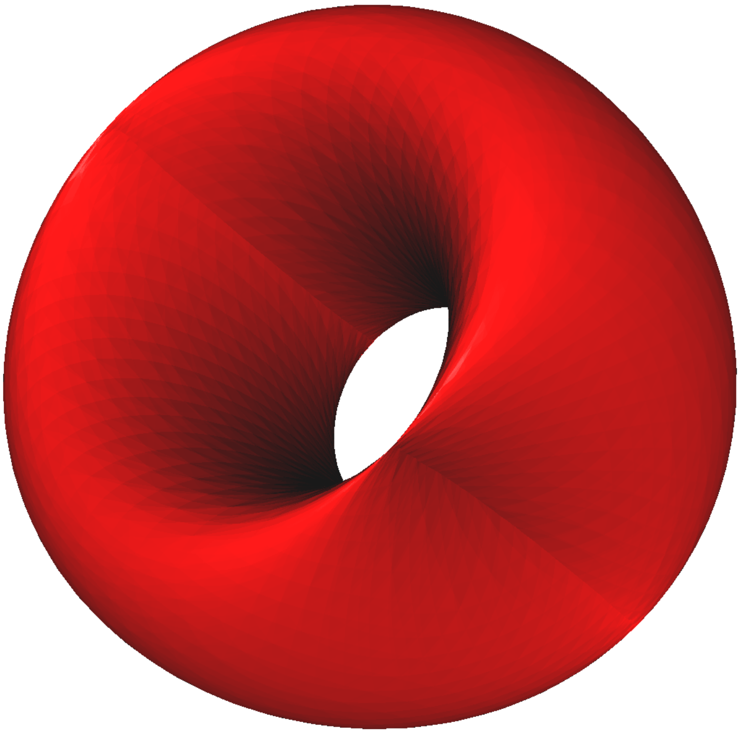

Figure 7 The piecewise-linear approximation of a flat torus embedded in $\mathbb{R}^{10}$ defined by the equations $x_1^2 + x_2^2 = 1$ and $x_3^2 + x_4^2 = 1$ and $x_i = 0$ for $i > 4$, projected to $\mathbb{R}^3$. The ambient triangulation used is a Coxeter triangulation of type $\tilde{A}_{10}$ with the diameter of the full-dimensional simplices 0.23. The output size $|\mathcal{S}|$ is 509952. The execution time of the algorithm is 231s.

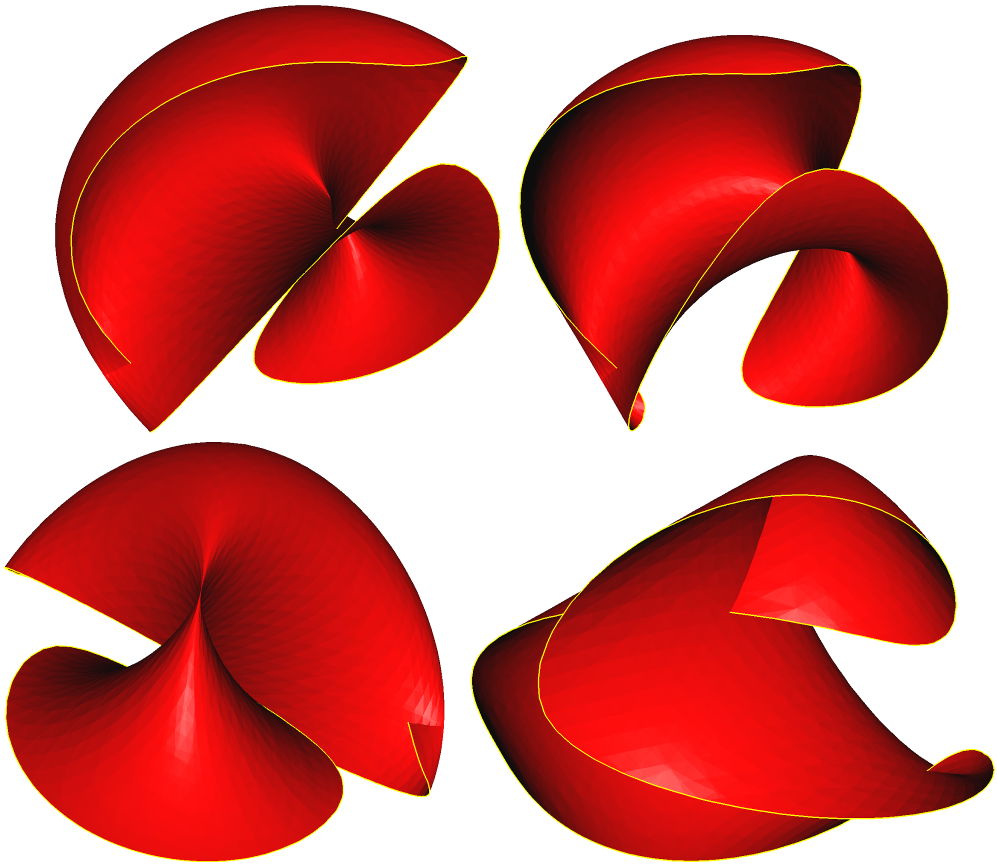

Figure 8 Four views of the flat torus in $\mathbb{R}^4$ given by two equations $x_1^2+x_2^2 = 1$ and $x_3^2+x_4^2 = 1$ cut by the hypersphere $(x_1-1)^2+x_2^2+(x_3-1)^2+x_4^2 = 4$, projected to $\mathbb{R}^3$. The ambient triangulation used is a Coxeter triangulation of type $\tilde{A}_4$ with the diameter 0.15 of the full-dimensional simplices. The reconstructed boundary is highlighted in yellow. The size $|\mathcal{S}|$ of the piecewise-linear approximation is 14779. The execution time of the algorithm is 1.84s.

References

1. Eugene Allgower and Kurt Georg. *Numerical continuation methods: an introduction*, volume 13. Springer Science & Business Media, 1990.
2. Eugene Allgower and Phillip H. Schmidt. An algorithm for piecewise-linear approximation of an implicitly defined manifold. *SIAM journal on numerical analysis*, 22(2):322–346, 1985.
3. Luigi Ambrosio and Halil Mete Soner. Level set approach to mean curvature flow in arbitrary codimension. *J. Differential Geom.*, 43(4):693–737, 1996. URL: `https://doi.org/10.4310/jdg/1214458529`, `doi:10.4310/jdg/1214458529`.
4. M. A. Armstrong. *Groups and Symmetry*. Springer, 1988.
5. Praveen Bhaniramka, Rephael Wenger, and Roger Crawfis. Isosurfacing in higher dimensions. In *Proceedings of the conference on Visualization'00*, pages 267–273. IEEE Computer Society Press, 2000.
6. Jean-Daniel Boissonnat and Arijit Ghosh. Manifold reconstruction using tangential Delaunay complexes. *Discrete & Computational Geometry*, 51(1):221–267, 2014.
7. Jean-Daniel Boissonnat, Leonidas J. Guibas, and Steve Oudot. Manifold reconstruction in arbitrary dimensions using witness complexes. *Discrete & Computational Geometry*, 42(1):37–70, 2009. URL: `https://doi.org/10.1007/s00454-009-9175-1`, `doi:10.1007/s00454-009-9175-1`.
8. Jean-Daniel Boissonnat, Siargey Kachanovich, and Mathijs Wintraecken. Triangulating submanifolds: An elementary and quantified version of Whitney's method. Preprint, December 2018. URL: `https://hal.inria.fr/hal-01950149`.
9. Jean-Daniel Boissonnat, Siargey Kachanovich, and Mathijs Wintraecken. Tracing isomanifolds of arbitrary dimension and codimension. In preparation, December 2019.
10. Jean-Daniel Boissonnat and Mathijs Wintraecken. The topological correctness of PL-approximations of isomanifolds. Preprint, November 2019. URL: `https://hal.inria.fr/hal-02386193`.
11. Nicolas Bourbaki. Lie groups and Lie algebras. Chapters 4–6. Translated from the 1968 French original by Andrew Pressley. Elements of Mathematics, 2002.
12. S. S. Cairns. On the triangulation of regular loci. *Annals of Mathematics. Second Series*, 35(3):579–587, 1934. `doi:10.2307/1968752`.
13. S-W. Cheng, T. K. Dey, and E. A. Ramos. Manifold Reconstruction from Point Samples. In *Proc. ACM-SIAM Symp. Discrete Algorithms*, pages 1018–1027, 2005.
14. Siu-Wing Cheng, Tamal K. Dey, and Jonathan Richard Shewchuk. *Delaunay Mesh Generation*. Chapman and Hall / CRC computer and information science series. CRC Press, 2013. URL: `http://www.crcpress.com/product/isbn/9781584887300`.
15. Aruni Choudhary, Siargey Kachanovich, and Mathijs Wintraecken. Coxeter triangulations have good quality. Submitted, December 2017. URL: `https://hal.inria.fr/hal-01667404`.
16. Kenneth L. Clarkson. Tighter bounds for random projections of manifolds. In *Proceedings of the 24th ACM Symposium on Computational Geometry, College Park, MD, USA, June 9-11, 2008*, pages 39–48, 2008. URL: `https://doi.org/10.1145/1377676.1377685`, `doi:10.1145/1377676.1377685`.
17. J. H. Conway and N. J. A. Sloane. *Sphere-packings, Lattices, and Groups*. Springer-Verlag New York, Inc., New York, NY, USA, 1987.
18. Harold S. M. Coxeter. Discrete groups generated by reflections. *Annals of Mathematics*, pages 588–621, 1934.
19. Wolfgang A. Dahmen and Charles A. Micchelli. On the linear independence of multivariate B-splines, I. Triangulations of simploids. *SIAM Journal on Numerical Analysis*, 19(5):993–1012, 1982.

20 Persi Diaconis, Susan Holmesand, and Mehrdad Shahshahani. Sampling from a manifold. *IMS Collections*, 2012.

21 David P. Dobkin, Allan R. Wilks, Silvio V. F. Levy, and William P. Thurston. Contour tracing by piecewise linear approximations. *ACM Transactions on Graphics (TOG)*, 9(4):389–423, 1990.

22 B. Curtis Eaves. *A course in triangulations for solving equations with deformations*, volume 234. Lecture Notes in Economics and Mathematical Systems, 1984.

23 Armin Eftekhari and Michael B. Wakin. What happens to a manifold under a bi-lipschitz map? *Discrete & Computational Geometry*, 57(3):641–673, 2017. URL: `https://doi.org/10.1007/s00454-016-9847-6`, `doi:10.1007/s00454-016-9847-6`.

24 Gideon Ehrlich. Loopless algorithms for generating permutations, combinations, and other combinatorial configurations. *Journal of the ACM (JACM)*, 20(3):500–513, 1973.

25 Hans Freudenthal. Simplizialzerlegungen von beschrankter flachheit. *Annals of Mathematics*, pages 580–582, 1942.

26 Alfred Gray. Comparison theorems for the volumes of tubes as generalizations of the weyl tube formula. *Topology*, 21(2):201–228, 1982.

27 GUDHI Project. GUDHI Editorial Board. URL: `http://gudhi.gforge.inria.fr/doc/latest/`.

28 George Haller and Sten Ponsioen. Nonlinear normal modes and spectral submanifolds: existence, uniqueness and use in model reduction. *Nonlinear dynamics*, 86(3):1493–1534, 2016.

29 James E. Humphreys. *Reflection groups and Coxeter groups*, volume 29. Cambridge university press, 1992.

30 Siargey Kachanovich. *Manifold meshing using Coxeter triangulations*. PhD thesis, Université Côte d'Azur, 2019.

31 S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006. Available at http://planning.cs.uiuc.edu/.

32 William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *ACM siggraph computer graphics*, volume 21, pages 163–169. ACM, 1987.

33 M. Maes and B. Kappen. On the permutahedron and the quadratic placement problem. *Philips Journal of Research*, 46(6):267–292, 1992.

34 Shawn Martin, Aidan Thompson, Evangelos A Coutsias, and Jean-Paul Watson. Topology of cyclo-octane energy landscape. *The journal of chemical physics*, 132(23):234115, 2010.

35 Chohong Min. Simplicial isosurfacing in arbitrary dimension and codimension. *Journal of Computational Physics*, 190(1):295–310, 2003.

36 Hariharan Narayanan and Partha Niyogi. Sampling hypersurfaces through diffusion. In *12th Intl. Workshop on Randomization and Computation (RANDOM)*, 2008.

37 B.C. Rennie and A.J. Dobson. On Stirling numbers of the second kind. *Journal of Combinatorial Theory*, 7(2):116 – 121, 1969. URL: `http://www.sciencedirect.com/science/article/pii/S0021980069800451`, `doi:https://doi.org/10.1016/S0021-9800(69)80045-1`.

38 Frank Ruskey and Carla D. Savage. Gray codes for set partitions and restricted growth tails. *Australasian J. Combinatorics*, 10:85–96, 1994.

39 Michael J. Todd. *The computation of fixed points and applications*, volume 124. Lecture Notes in Economics and Mathematical Systems, 1976.

40 Nakul Verma. A note on random projections for preserving paths on a manifold. Technical Report Tech. Report CS2011-0971, UC San Diego, 2011.

41 Timothy R. Walsh. Loop-free sequencing of bounded integer compositions. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 33:323–345, 2000.

42 Rephael Wenger. *Isosurfaces: geometry, topology, and algorithms.* AK Peters/CRC Press, 2013.
43 Hermann Weyl. On the volume of tubes. *American Journal of Mathematics*, 61(2):461–472, 1939.
44 J. H. C. Whitehead. On C^1-complexes. *Annals of Mathematics*, 41(4):809–824, 1940. URL: `http://www.jstor.org/stable/1968861`.
45 H. Whitney. *Geometric Integration Theory.* Princeton University Press, 1957.
46 G. M. Ziegler. *Lectures on Polytopes.* Graduate Texts in Mathematics. Springer New York, 2012. URL: `https://books.google.fr/books?id=xd25TXSSUcgC`.

A Proofs for Section 2.4

In this appendix we will prove that the Freudenthal-Kuhn triangulation is the Coxeter triangulations of type $\tilde{A}_d$ up to a linear transformation. We also prove that the Voronoi cell of a vertex in a Coxeter triangulation of type $\tilde{A}_d$ is a permutahedron.

For this we need to first recall an equivalent definition of the Coxeter triangulations of type $\tilde{A}_d$: Any Coxeter triangulation can be defined as an hyperplane arrangement

$$\mathcal{H} = \{H_{r,k} \mid r \in R_+, k \in \mathbb{Z}\},$$

where

$$H_{u,k} = \{x \in \mathbb{R}^d \mid \langle x, u \rangle\},$$

and R_+ denotes the set of *positive roots* of the Coxeter group. We will not recall the general definition of positive roots, which can be found in for example [11, 29, 15], but use that for $\tilde{A}_d$, according to [11, Planche II],

$$R_+ = \left\{ r_{i,j} = \sum_{l=i}^{j} s_l \,\middle|\, 1 \le i \le j \le d \right\},$$

with $\{s_l\}$ the *simple roots* of the Weyl group A_d associated to the triangulation of type $\tilde{A}_d$. For a discussion of the Weyl group we again refer to for example [11, 29, 15]. We also recall the simple roots of A_d, which will be needed in the second proof. The simple roots of A_d (in the hyperplane $P = \{(x^i) \in \mathbb{R}^{d+1} \mid \sum_i x^i = 0\} \subset \mathbb{R}^{d+1}$) are in turn:

$$s_1 = e_1 - e_2,\ s_2 = e_2 - e_3,\ \ldots,\ s_d = e_d - e_{d+1},$$

see [11]. We stress that these simple roots can be rescaled, permuted and rotated in the hyperplane P. We note that one can easily rotate $\mathbb{R}^d \subset \mathbb{R}^{d+1}$ given by the first d basis vectors into P. The matrix in $\mathrm{SO}(d+1)$ of this transformation has as first d rows $((1/\sqrt{i^2+i})^{\{i\}}, -i/\sqrt{i^2+i}, 0^{\{d-i-1\}})$ and the final row $((1/\sqrt{d+1})^{\{d+1\}})$, where $c^{\{k\}}$ denotes k consecutive coordinates equal to c. We will not use this transformation because it complicates the expressions prohibitively.

In Lemma 9, we have seen that $x \in \mathbb{R}^d$ lies on the face of some simplex with canonical permutahedral representation $(\tilde{v}_0, \omega)$ in the FK-triangulation if and only if either $x^i - \tilde{v}_0^i = x^j - \tilde{v}_0^j$ or $x^i - \tilde{v}_0^i = 0$ for some i, j. Note that $\tilde{v}_0^i, \tilde{v}_0^j \in \mathbb{Z}$. Hence we see that

▶ **Lemma 27.** *The Freudenthal-Kuhn triangulation is a hyperplane arrangement $\tilde{\mathcal{H}} = \{H_{u,k} \mid u \in E, k \in \mathbb{Z}\}$, with*

$$E = \{e_1, \ldots, e_d\} \cup \{u_{i,j} = e_j - e_i \mid 1 \le i < j \le d\}.$$

We now define a linear map μ from $\mathbb{R}^d$ to P by showing how it acts on the basis:

$$\mu(e_i) = r_{1,i} = \sum_{i=1}^{j} s_i.$$

We claim that μ maps E bijectively onto R_+. The vector $\mu(e_i) = r_{1,i}$ lies in R_+, by construction. For $u_{i,j} \in E$, with $i < j$, we see that

$$\mu(u_{i,j}) = \mu(e_j - e_i) = \mu(e_j) - \mu(e_i) = r_{1,j} - r_{1,i} = \sum_{l=1}^{j} s_l - \sum_{l=1}^{i} s_l = \sum_{l=i+1}^{j} s_l = r_{i+1,j}.$$

Hence $\mu(u_{i,j})$ lies in R_+. By reading the previous calculation backwards we see that μ^{-1} maps each $r \in R_+$ to a vector in E.

We conclude that μ (bijectively) maps $\mathcal{H}$ to $\tilde{\mathcal{H}}$, which completes the proof of Lemma 19.

We now prove the following:

▶ **Proposition 28.** *The Voronoi cell of a Coxeter triangulation of type $\tilde{A}_d$ is a permutahedron.*

Proof. We start by recalling a number of results. In [15] we have seen that the circumcentre of the simplex given in Definition 18 is

$$c = \left(-\frac{d-2i}{2(d+1)} \right),$$

with $i \in \{0, \dots, d\}$. The circumcentre of a Delaunay simplex is a Voronoi vertex. We recall that

- All simplices in the star of 0 in the Coxeter triangulation are found by consecutive reflection of the simplex of Definition 18 in the hyperplanes of $\mathcal{H}$ that go through 0, that is the hyperplanes with normals $r_{j,k} = e_j - e_k$, with $j \neq k$. See for example [11, 29, 15]. We also call these reflections the action of the Weyl group.
- The reflection $R_{j,k}$ in a plane that goes through the origin with normal $r_{j,k}$ is given by
$$R_{j,k}(v) = v - 2\frac{v \cdot r_{j,k}}{r_{j,k} \cdot r_{j,k}} r_{j,k} = v - (v \cdot r_{j,k}) r_{j,k}.$$

We find that

$$R_{j,k}\left(c\right)^i = \left(c - (c \cdot r_{j,k}) r_{j,k}\right)^i = -\frac{d-2i}{2(d+1)} - \frac{2j-2k}{2(d+1)}(\delta_{ij} - \delta_{ik}),$$

which permutes the jth and kth coordinate of c. Here we used the upper index i to denote the ith coordinate. Using the cycle notation for the permutation group, see for example [4, Chapter 6], this coincides the 2-cycle $(j\,k)$. Let now

$$c_\pi = \left(-\frac{d-2\pi_i}{2(d+1)} \right),$$

with $\{\pi_i\}$ some permutation of $\{0, \dots, d\}$. We find that

$$R_{j,k}\left(c_\pi\right)^i = \left(c_\pi - (c_\pi \cdot r_{j,k}) r_{j,k}\right)^i = -\frac{d-2\pi_i}{2(d+1)} - \frac{2\pi_j - 2\pi_k}{2(d+1)}(\delta_{ij} - \delta_{ik}),$$

which again permutes the jth and kth coordinate. Now recall that all permutations are generated by 2-cycles, see for example [4, Theorem 6.1]. This implies that, for any permutation π, we can find c_π from c by the action of the Weyl group. This also means that we have explicitly described the Voronoi cell of 0 in the Coxeter triangulation of type $\tilde{A}_d$ as a permutahedron. Because of symmetry, this now holds for any Voronoi cell. ◀

B Proofs for Section 2.1

The proof of Corollary 6 is based on:

▶ **Lemma 29** (Lemma 3.11 of [33]). *The face of a permutahedron corresponding to an ordered partition $\omega = (\omega_1, \dots, \omega_{l+1})$ is combinatorially*

$$\mathcal{P}(|\omega_1|) \times \cdots \times \mathcal{P}(|\omega_{l+1}|),$$

where $|\omega_i|$ denotes the length of the ith part of the ordered partition and $\mathcal{P}(k)$ the permutahedron of dimension k.

Proof of Corollary 6. Since the number of vertices of the product of two polytopes is the product of the vertices and a k dimensional permutadedron has $(k+1)!$ vertices, we see that the total number of vertices of a face of a permutahedron corresponding to an ordered partition $\omega = (\omega_1, \dots, \omega_{l+1})$ is

$$\prod_i (|\omega_i|!).$$

Let $1 \leq k < j \leq d$, be integers such that $k+j = d+1$. By definition $k!j! < (k-1)!(j+1)!$, and thus $k!j! \leq 1!d!$. Generalizing this, we see that the product of the $|\omega_i|!$ is maximal when all parts are singletons except the biggest part which has $d+1-l$ elements. Therefore

$$\prod_i (|\omega_i|!) \leq (d-l+1)!.$$

◀

Proof of Lemma 7. We first recall a set of $d > 2$ objects can be subdivided in two non-empty ordered subsets A and B in $2^d - 2$ ways. This is not hard to see. Because we pick for each element if it will be put in A or B there are 2^d possibilities. Excluding that A or B is empty gives $2^d - 2$. Let $\omega = (\omega_1, \dots, \omega_l)$ again be an ordered partition. To find a refinement of ω in $l+1$ parts we need to first pick a $1 \leq i \leq l$, such that $|\omega_i| > 1$ and then we need to break ω_i up into two (ordered) parts, for which there are $2^{|\omega_i|} - 2$ possibilities as we have seen above. This means that if $I = \{i \mid 1 \leq i \leq l, |\omega_i| > 1\}$, the number of refinements is

$$\sum_{i \in I} 2^{|\omega_i|} - 2.$$

Let now $1 \leq k < j \leq d$ be integers such that $k + j = d + 1$. Then $2^k + 2^j < 2^{k-1} + 2^{j+1}$. Generalizing this, we see that the sum of the $2^{|\omega_i|} - 2$ is maximal when all $|\omega_i| = 1$ except the biggest part which has $d - l + 1$ elements. ◀